Visual 3D scene reconstruction

From dense stereo to single camera structure from motion

Roland Brockers

4/22/2010

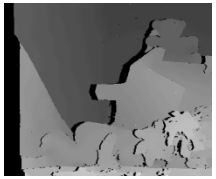
Introduction

Former projects:

University of Paderborn / Germany
Labor für Bildverarbeitung

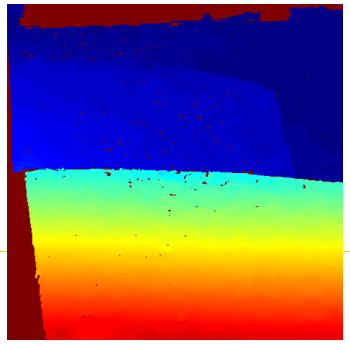
- Biologically inspired computer vision algorithms
  - Object recognition, 3D reconstruction
- Mobile robot platform design for tele-presence applications
- Immersive human machine interfaces
- Virtual Robot Simulator

PhD: Cooperative Stereo Vision Algorithm using Cost Relaxation



DFG research grant (at JPL):
Stereo vision for mobile robot platforms

- Accurate 3D object borders
- Speed-up optimization algorithm
- Temporal stereo processing



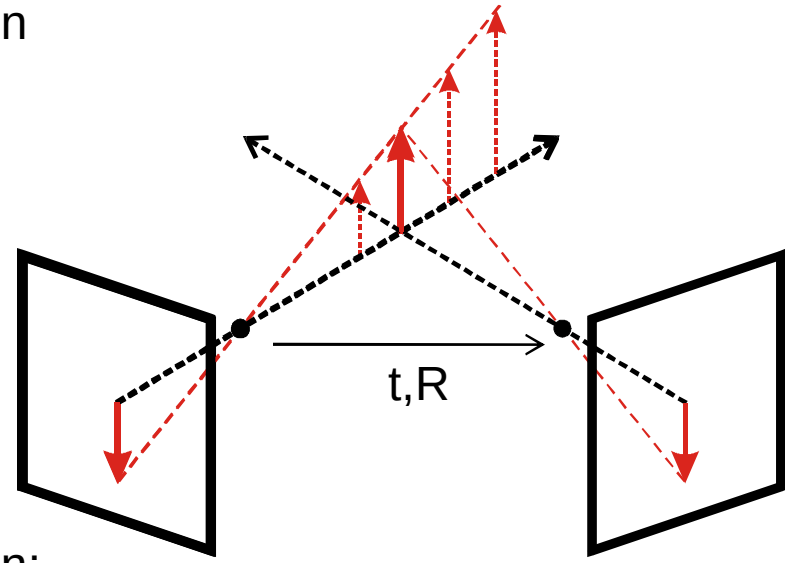
Overview

- Two view geometry and 3D scene reconstruction
- Stereo Vision
- Single Camera Structure from Motion
- Summary & Conclusion

3D scene reconstruction

3D reconstruction using cameras as passive sensors

- 2D projection of 3D world
- At least 2 views for reconstruction
- Reconstruction relative to translation $\mathbf{t}$



- two cameras in fixed constellation:
  $\mathbf{t}$, $\mathbf{R}$ known → stereo vision
- single moving camera:
  $\mathbf{t}$, $\mathbf{R}$ unknown → structure from motion up to scale

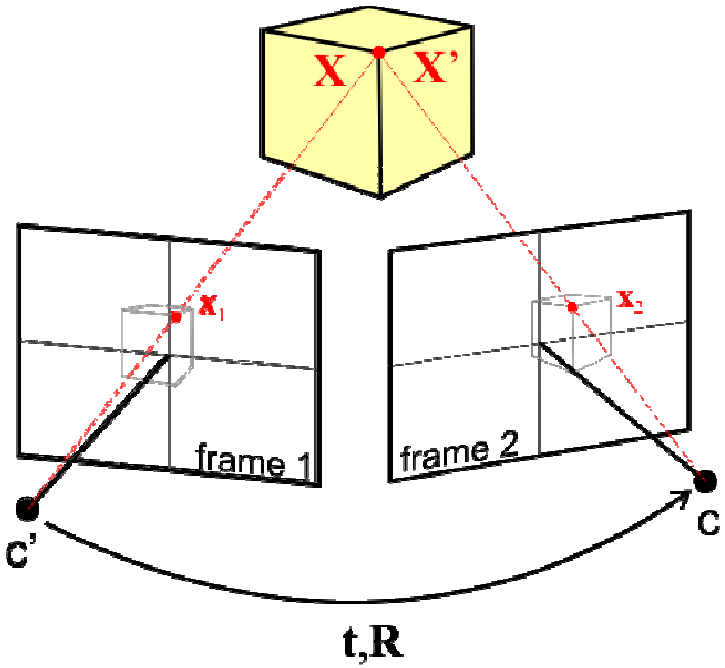
Two view geometry

Perspective projection

- 3D points $\mathbf{X} = [X, Y, Z, W]^T \in \Re^4, \quad (W = 1)$
- Image points $\mathbf{x} = [x, y, z]^T \in \Re^3, \quad (z = 1)$
- Perspective projection

$$\lambda \mathbf{x} = \mathbf{X}$$

$$\lambda = Z, \quad x = \frac{X}{Z}, \quad y = \frac{Y}{Z}$$



- Rigid body motion $\mathbf{X} = \mathbf{R}\mathbf{X}' + \mathbf{t}$
- Rigid body motion + projective projection $\Pi = [\mathbf{R}, \mathbf{t}] \in \Re^{3x4}$

$$\lambda \mathbf{x} = \Pi \mathbf{X}' = [\mathbf{R}, \mathbf{t}]\mathbf{X}'$$

$$\lambda_2 \mathbf{x}_2 = \mathbf{R}\lambda_1 \mathbf{x}_1 + \mathbf{t}$$

Epipolar geometry

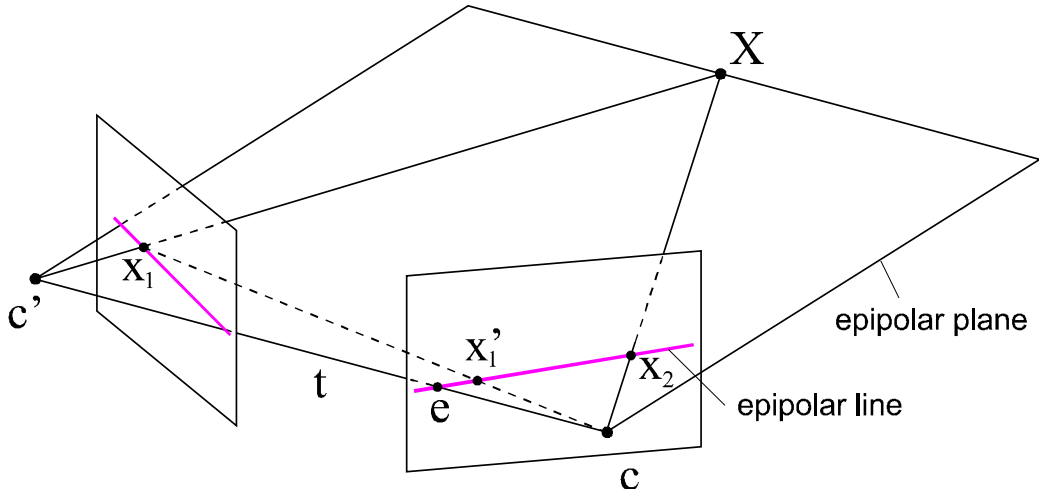
Epipolar constraint:

$\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{t}$ are coplanar

$$\overrightarrow{\mathbf{c}'\mathbf{x}_1} \cdot [\overrightarrow{\mathbf{c}'\mathbf{c}} \times (\overrightarrow{\mathbf{c}\mathbf{x}_2})] = 0$$

$$\mathbf{x}_1^T \cdot [\mathbf{t} \times (\mathbf{R}\mathbf{x}_2)] = 0$$

$$\mathbf{t} \times \mathbf{R}\mathbf{x}_2 = [\mathbf{t}_\times]\mathbf{R}\mathbf{x}_2$$



$\mathbf{E} = [\mathbf{t}_\times]\mathbf{R}$ Essential matrix
[Longuet-Higgings 1981]

$\mathbf{x}_1^T \mathbf{E}\mathbf{x}_2 = 0$

- Maps a point $\mathbf{x}_1$ to an epipolar line $\mathbf{E}\mathbf{x}_2$
- 5 independent parameters (up to scale)
- assumes intrinsic parameters are known

Stereo vision: $\mathbf{t}$, $\mathbf{R}$ fixed and known from calibration

→ epipolar geometry can be pre-calculated

→ use rectification + epipolar constraint for correspondence search

Single moving camera: $\mathbf{t}$, $\mathbf{R}$ unknown

→ prior to 3D reconstruction, do motion estimation

Overview

- Two view geometry and 3D scene reconstruction
- Stereo Vision
- Single Camera Structure from Motion
- Summary & Conclusion

Stereo vision

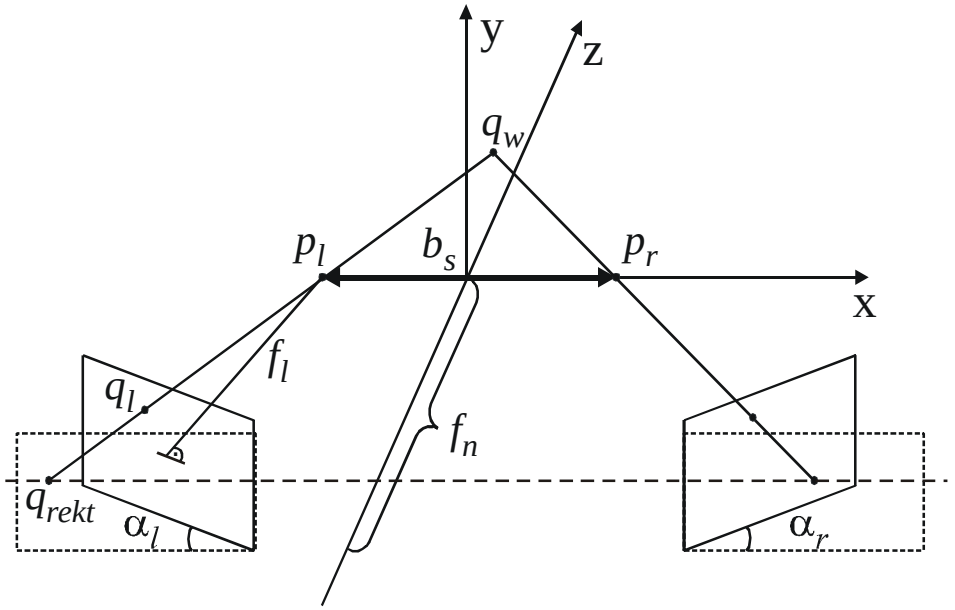
Image processing pipeline

- Image acquisition & Pre-processing
- Correspondence search
- Passive triangulation

Stereo vision

Image processing pipeline

- Image acquisition & Pre-processing
  - Image correction (e.g. removing of lens distortion)
  - Rectification

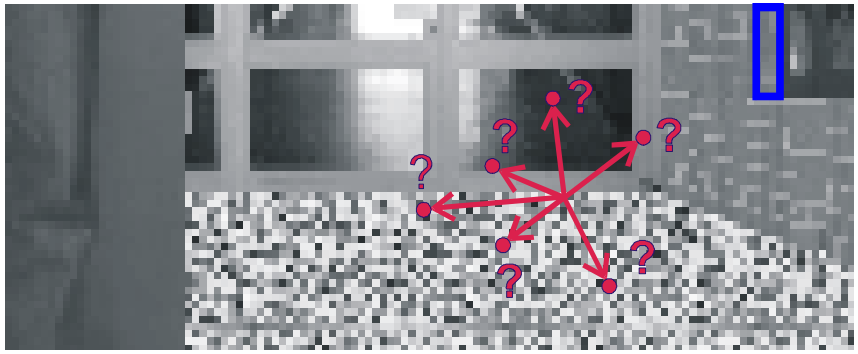


Stereo vision

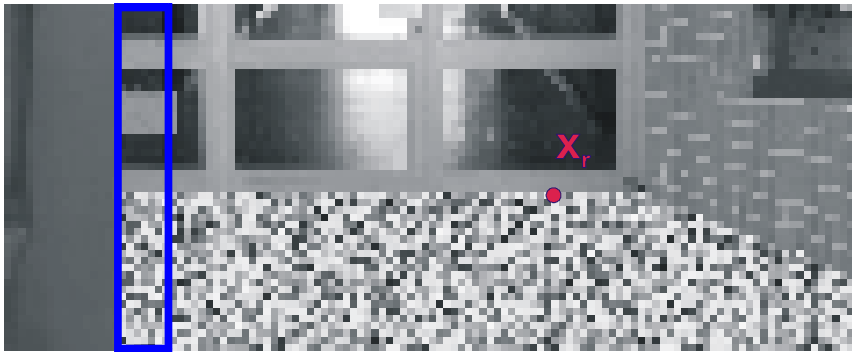
Image processing pipeline

- Image acquisition & Pre-processing
- Correspondence search
  - Calculation of a disparity map
  - Difficulties due to ambiguities or occlusions

Occlusions



left view

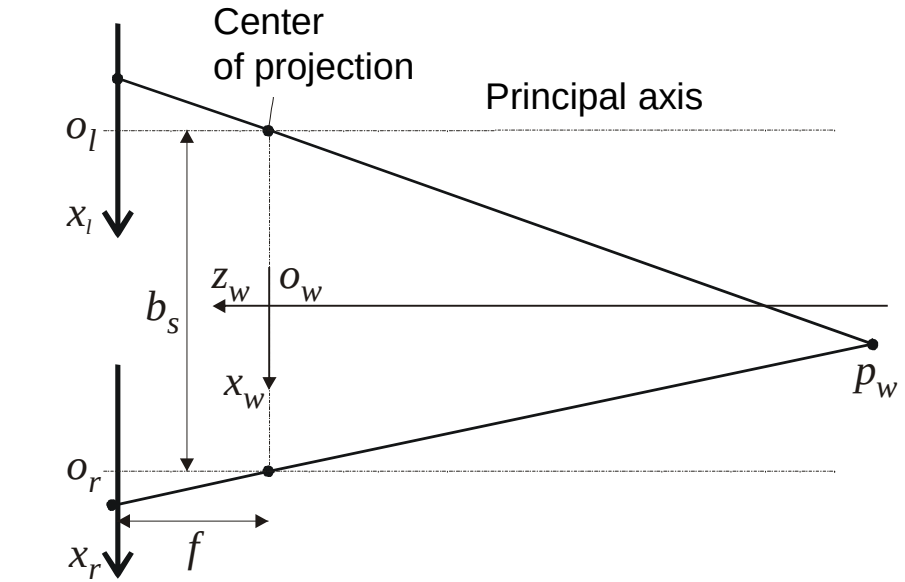


right view

Stereo vision

Image processing pipeline

- Image acquisition & Pre-processing
- Correspondence search
- Passive triangulation

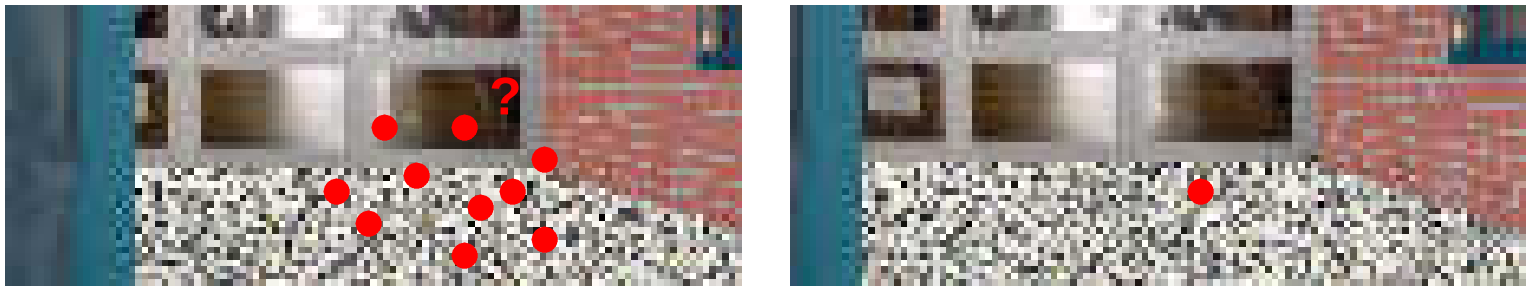


Disparity: $d_x = x_l - x_r$

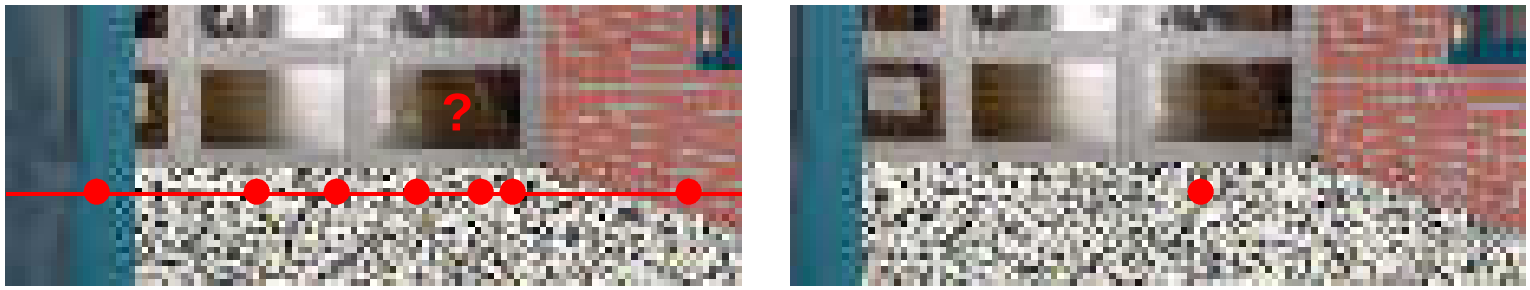
Range: $x_w = \frac{b_s(x_l + x_r)}{2d_x}$

$z_w = -\frac{f\, b_s}{d_x}$

Correspondence search

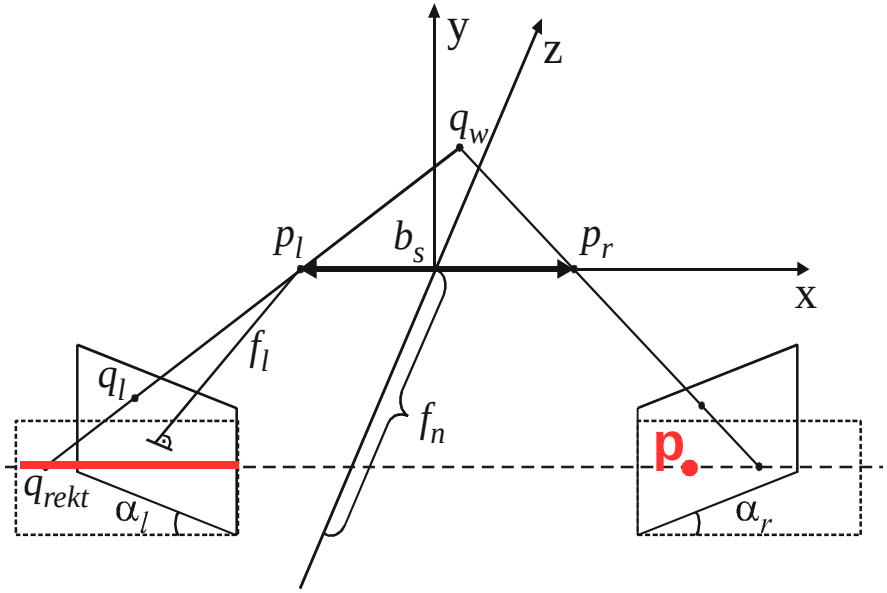


Correspondence search

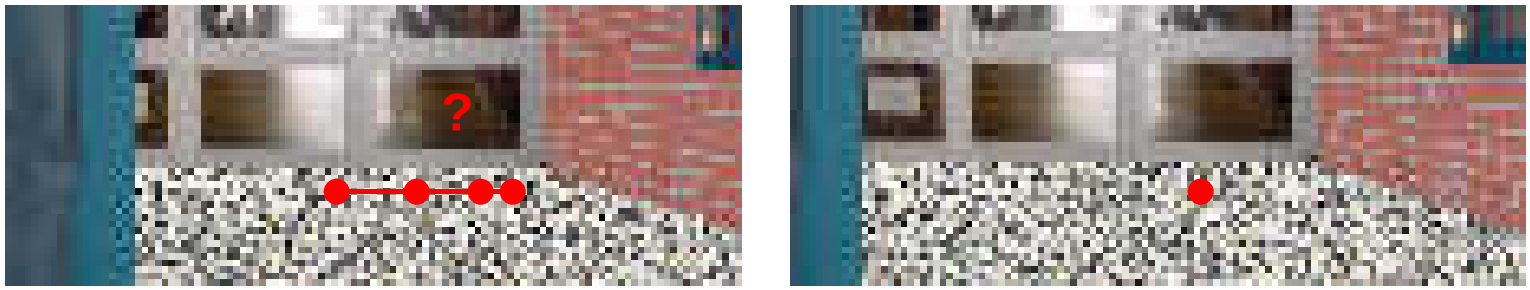


Stereoscopic Constraints

- Epipolar constraint



Correspondence search



Stereoscopic Constraints

- Epipolar constraint
- Limited disparity range

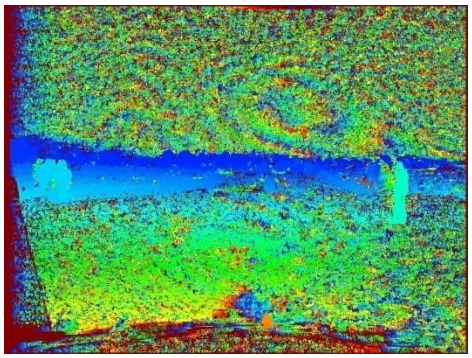
- No transparent objects: Uniqueness constraint [Marr & Poggio, 1976]
- Solid objects: Continuity constraint [Marr & Poggio, 1976]

- Disparity gradient limit (humans: < 1.0) [Burt & Julesz, 1980]
- Order constraint

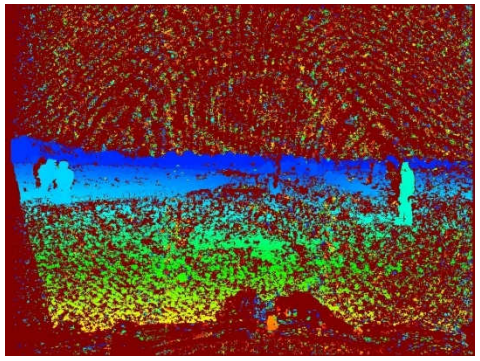
SAD1 correspondence search

- Real time stereo algorithm
- Correlation with sum of absolute differences (SAD)

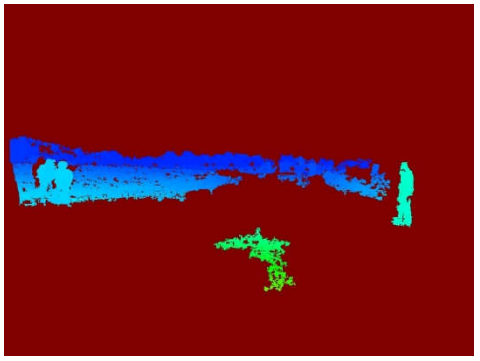
- Result, SAD1
  7x7 correlation windows



- → Post filter needed



Left-right consistency check



Blob filter

Motivation for Costrelax dense stereo algorithm

Motivation:

- Biologically inspired cooperative approach
- Optimizing of disparities by coupling the correspondence problems of neighboring image pixels
- Acceleration of the optimization by reformulating the optimization problem as a cost minimization

Implementation:

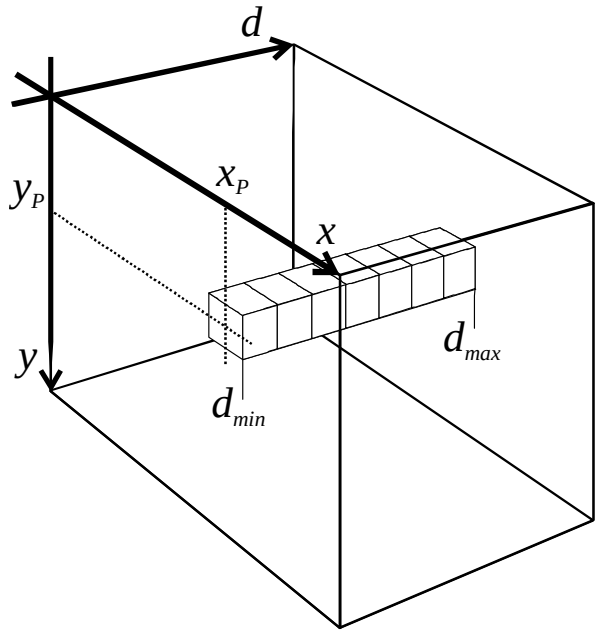
- Utilization of stereoscopic continuity constraint
- Global cost function with unique minimum
- Fast, iterative determination of the minimum with standard procedure, e. g. gradient descend algorithm
- On interruption of the iteration, calculation of a „momentary“ disparity map, providing the best possible solution at the moment of interruption

Costrelax algorithm

1. Initial correlation (Normalized Cross Correlation, 3x3 corr. windows)
2. Iterative optimization of correlation scores
3. Explicit occlusion detection and subpixel refinement

Costrelax algorithm

Disparity space $s(x,y,d)$



Definition of variable vector ξ:

$$(x, y) \rightarrow k$$

$$s(x, y, d) \rightarrow \xi_{(k,d)} \qquad \text{with } k \in [1,...,n] \text{ and } d \in [d_{\min},...,d_{\max}]$$

$$\xi = (\xi_{(1,d_{\min})}, \ldots, \xi_{(n,d_{\min})}, \xi_{(1,d_{\min}+1)}, \ldots, \xi_{(n,d_{\max})})^T$$

Costrelax algorithm

Global cost function:

$$P(\xi) = c_1 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} (\xi_{(i,d)} - \xi_{(i,d)_0})^2 + c_2 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} \sum_{j \in U_i} w_{ij} (\xi_{(i,d)} - \xi_{(j,d)})^2$$

Distance to similarity measure

Continuity constraint

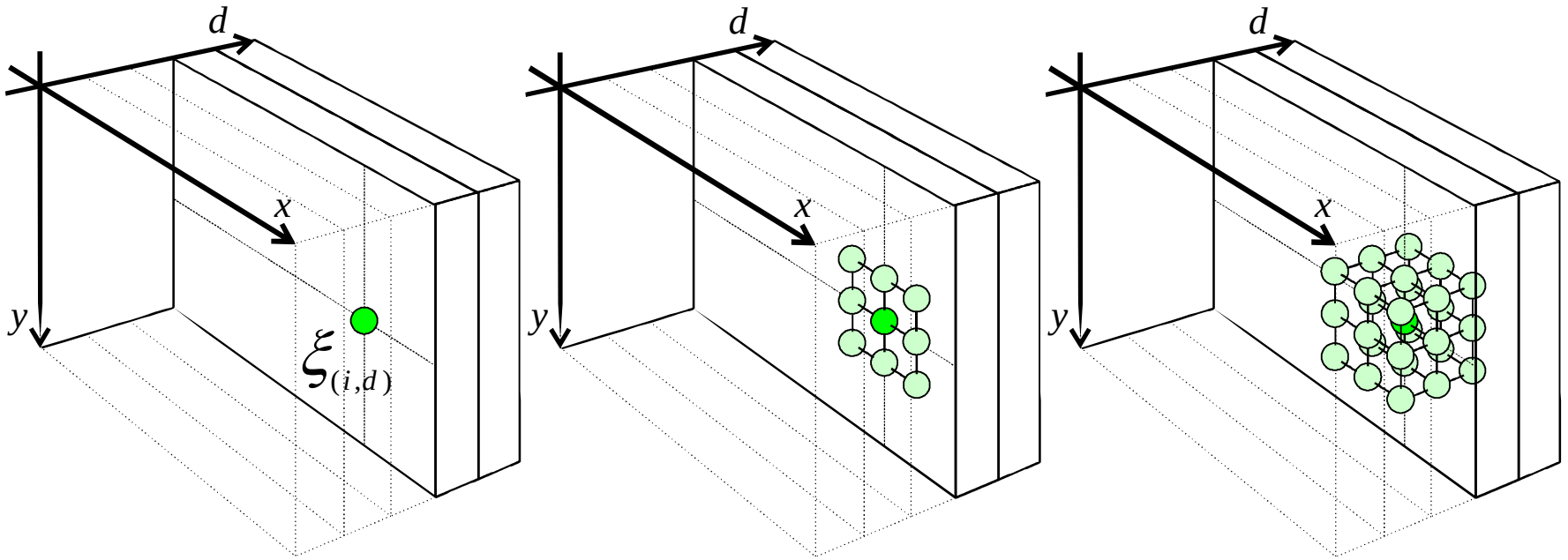
Coupling of neighboring variables in a local support area U_i.

Initial correlation scores: $s_0(x, y, d) \rightarrow \xi_{(k,d)_0}$

Costrelax algorithm

Local support area U_i:

- inside a constant disparity level (2D-window function)
- inside disparity space (3D-window function)



2D-support area

3D-support area

Interpretation of the coupled variable system as a neural network
[Marr & Poggio, 1976], [Reimann & Haken, 1994]

Costrelax algorithm

Finding the minimum of P

Necessary condition: $\nabla P(\xi) = 0$

with $$\frac{\partial P}{\partial \xi_{(i,d)}}(\xi) = (2c_1 + 4c_2 \sum_{j \in U_i} w_{ij})\xi_{(i,d)} - 2c_1 \xi_{(i,d)_0} - 4c_2 \sum_{j \in U_i} w_{ij} \xi_{(j,d)}$$

we get a linear equation system $\mathbf{A}\xi - \xi_0 = 0$ with a symmetric, positive definite matrix $\mathbf{A} \in R^{n \times n}$, that is: it exists an inverse $\mathbf{A}^{-1}$ with a well-defined solution.

Iteration with gradient decend algorithm:

$$\xi_{i+1} = \xi_i - \lambda \nabla P(\xi_i)$$

constant step width $$0 < \lambda < \frac{1}{c_1 + 4Qc_2}$$ with $Q = \sum_{j \in U_i} w_{ij}$

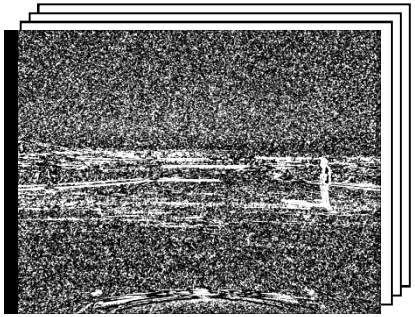
Costrelax algorithm

After convergence:

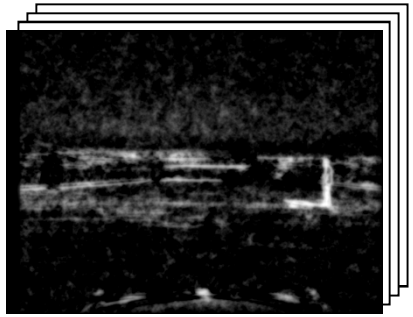
- Determination of the pixel disparity by searching the maximum value of all assigned variables

$$d(k) = j\Big|_{\xi(k,j)=\max\{\xi(k,i)\}} \quad \text{with} \quad i \in \{d_{\min}, ..., d_{\max}\}$$

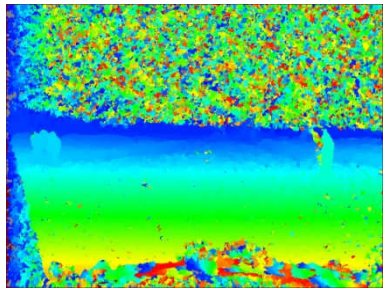
Example: Variable values for a fixed disparity of 36 pixels (right person disparity)



Before optimization



After optimization



Disparity map

Post processing

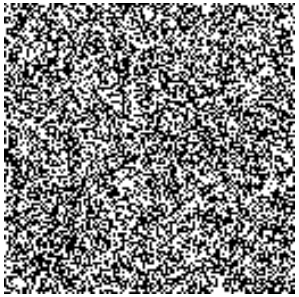
Explicit occlusion detection

$$d(k) = \begin{cases} d(k), & \xi_{(k,d(k))} = \max\{\xi_{(r,d(r))}\}\big|_{x_l = r + d(r)} \\ c, & \text{otherwise} \end{cases}$$

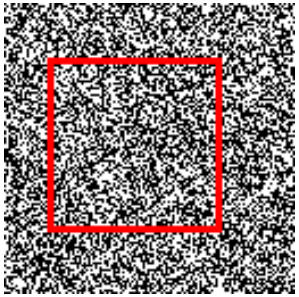
Maximum search across all image pixels corresponding with the same pixel in the other view.

(Uniqueness constraint)

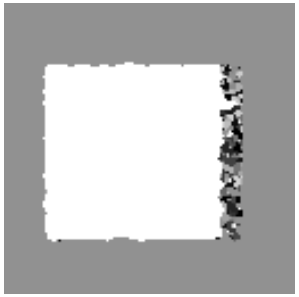
Example: Random dot stereogram with a shifted quadratical texture patch



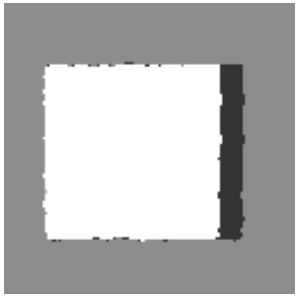
left view



right view



Result without occlusion detection



Result including occlusion detection

Post processing

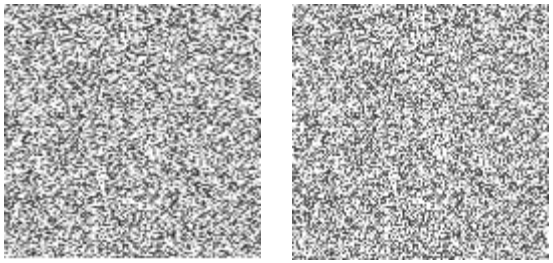
Sub-pixel refinement

Cost function

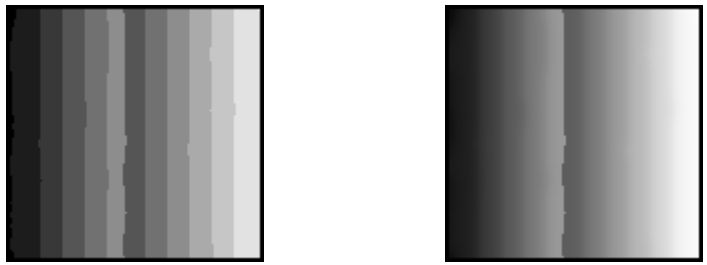
$$P(\tilde{\xi}) = c_3 \sum_{i=1}^{m} (d(i) - d_0(i))^2 + c_4 \sum_{i=1}^{m} \sum_{j \in \tilde{U}_i} (d(i) - d(j))^2$$

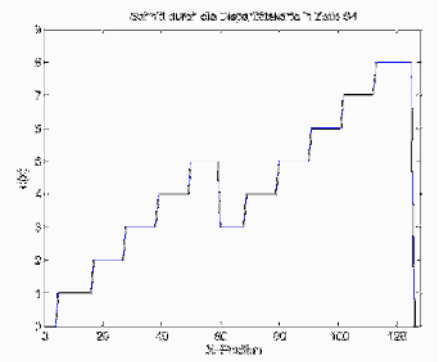
with $\tilde{\xi} = [d_1, \ldots, d_n]^T$

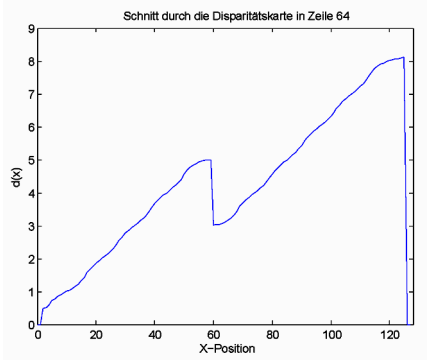
for all $|d(i) - d(j)| \leq 1$



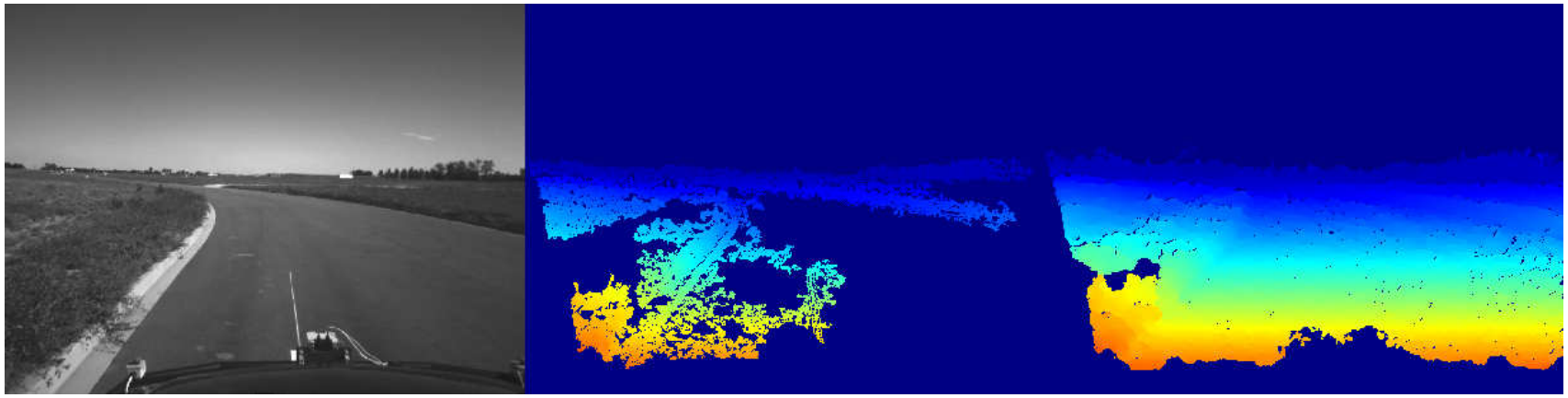
Random dot stereogram, with a 10% horizontal stretch of the left view and a 2 pixel shift in the middle of the image







First results



Original left view | Disparity map SAD5 stereo | Disparity map Costrelax

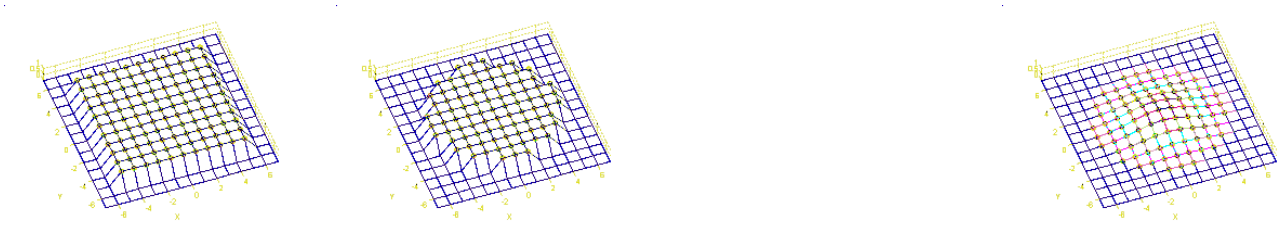
Local support

Local support area

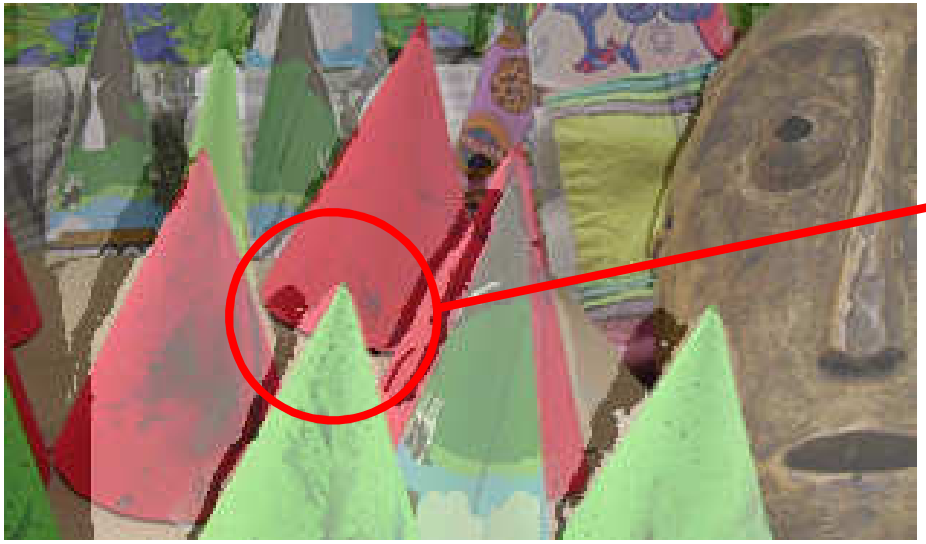
$$P(\xi) = c_1 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} (\xi_{(i,d)} - \xi_{(i,d)_0})^2 + c_2 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} \sum_{j \in U_i} w_{ij} (\xi_{(i,d)} - \xi_{(j,d)})^2$$

U_i defines the outer shape of local support for pixel i

w_{ij} defines the influence of neighbor j on i



Fixed local support



Foreground fattening effect
caused by fixed local support

Color-based adaptive weight local support


$$P(\xi) = c_1 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} (\xi_{(i,d)} - \xi_{(i,d)_0})^2 + c_2 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} \sum_{j \in U_i} \gamma_{ij} (\xi_{(i,d)} - \xi_{(j,d)})^2$$

Adaptive local support windows:

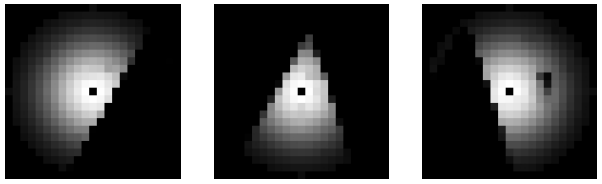
- fixed neighborhood window U_i
- weight factor γ_{ij} for each window pixel depending on Gaussian weighted color distance in CIE-Lab color space and the Euclidian distance to the window center

$$\gamma_{ij} = r_{ij} \cdot c_{ij} \qquad r_{ij} = e^{-\frac{1}{2}\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{\sigma_r^2}} = e^{-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\sigma_r^2}}$$

$$c_{ij} = e^{-\frac{1}{2}\frac{(\mathbf{c}_i - \mathbf{c}_j)^4}{\sigma_c^2}} = e^{-\frac{((L_i - L_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2)^2}{2\sigma_c^2}}$$

Color-based adaptive weight local support

Example: Adaptive weights support window

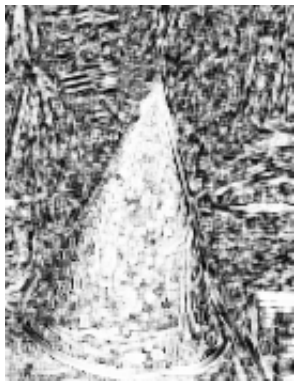


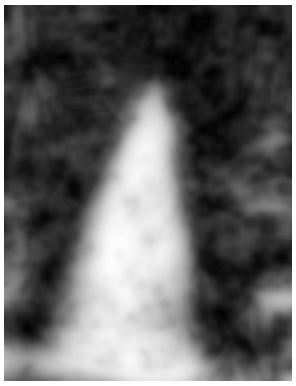
1 2 3

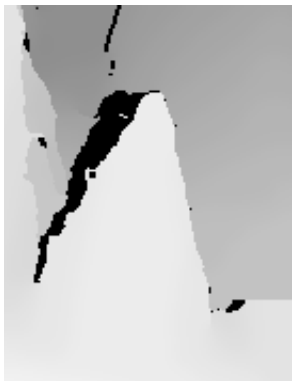
Support weights

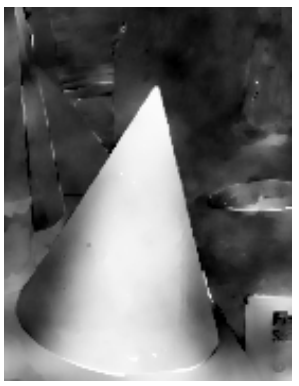
Fixed local support

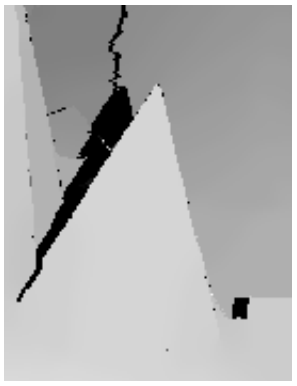
Adaptive local support





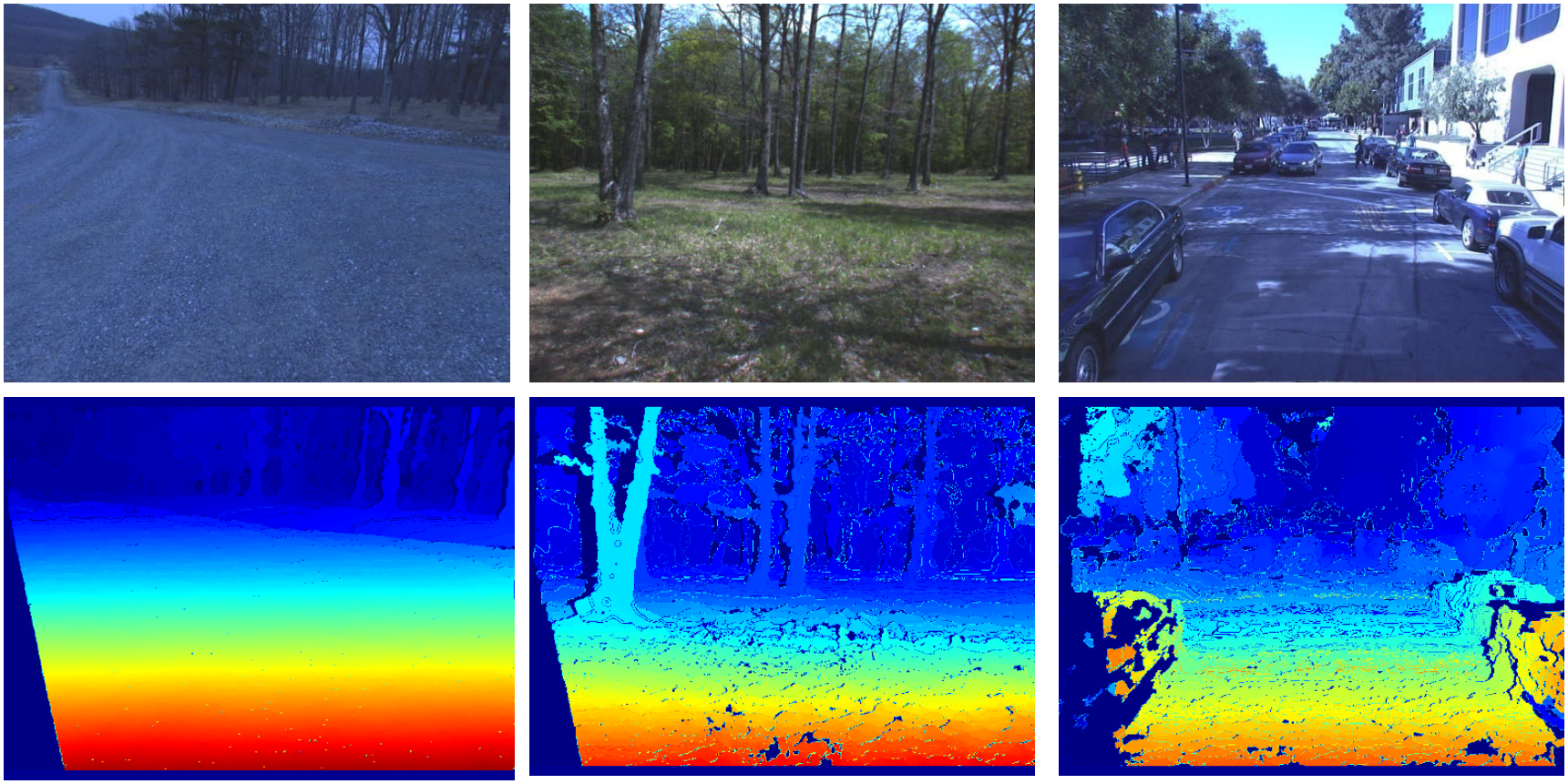






Qualitative Evaluation

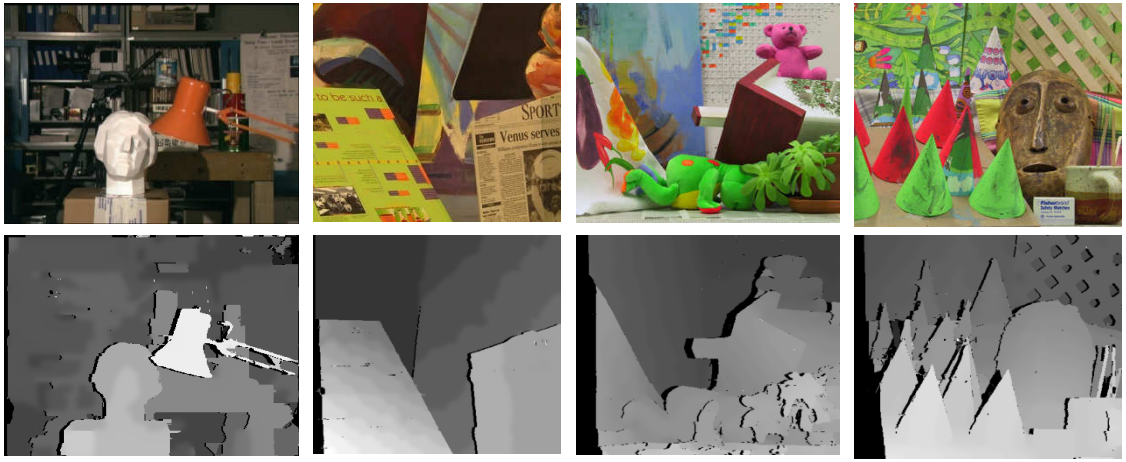
Costrelax with adaptive weight local support



Quantitative Evaluation

Percentage of bad matching pixels for the Middlebury data set ($\delta d = 1$) for non occluded pixels (non), all pixels (all) and near discontinuities (disc) (cp. http://vision.middlebury.edu/stereo)

Algorithm	Tsukuba			Venus			Teddy			Cones		
	non	all	disc	non	all	disc	non	all	disc	non	all	disc
Segment Support [1]	1.25	1.62	6.68	0.25	0.64	2.59	8.43	14.2	18.2	3.77	9.87	9.77
Adaptive Weight [2]	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	8.26
CostRelax [5] (Adapt. weight local support)	**2.91**	**3.49**	**11.4**	**0.60**	**1.11**	**6.45**	**7.92**	**13.7**	**20.9**	**3.59**	**9.43**	**10.3**
CostRelax [3] (3D fixed loc. supp.)	4.76	6.08	20.3	1.41	2.48	18.5	8.18	15.9	23.8	3.91	10.2	11.8
CostRelax [4] (2D fixed loc. supp.)	6.33			1.44			9.60			5.24		



[1] Tombari, F.,Mattoccia, S., Di Stefano, L.: Segmentation-based adaptive support for accurate stereo correspondence. PSIVT 2007. LNCS 4872, pp. 427–438 (2007)

[2] Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Trans. PAMI 28, 650–656 (2006)

[3] Brockers, R., Hund, M., Mertsching, B.: Stereo vision using cost-relaxation with 3d support regions. IVCNZ 2005, pp. 96–101 (2005)

[4] Brockers, R., Hund, M.,Mertsching, B.: Stereo matching with occlusion detection using cost relaxation. ICIP 2005, pp. 389–392 (2005)

[5] Brockers, R.: Cooperative Stereo Matching with Color-Based Adaptive Local Support. CAIP 2009, LNCS 5702, pp. 1019–1027 (2009)

Temporal stereo extension

In image sequences use previous results as a disparity prior

$$P(\xi) = c_1 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} (\xi_{(i,d)} - \xi_{(i,d)_0})^2 + c_2 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} \sum_{j \in U_i} w_{ij} (\xi_{(i,d)} - \xi_{(j,d)})^2$$

$$+ c_3 \sum_{d=d_{\min}}^{d_{\max}} \sum_{i=1}^{n} (d = d_{pr}(i)) \rho_i (\xi_{(i,d)} - 1)^2$$

New term generates costs for all variables with prior disparity d_{pr}, if variable value < 1

Several approaches to generate prior:

- Previous disparity map
- Previous 2 disparity maps + extrapolation
- Warped previous disparity map using motion estimation from Visual Odometrie

Table: Average improvement of false matching error

Prior	Forward motion		Side motion	
	No noise	SNR 30dB	No noise	SNR 30dB
Sub-sampled damp	9.53%	30.56%	-10.47%	18.52%
Previous dmap	31.04%	6.44%	2.46%	4.79%
Prev. dmap + confidence	8.83%	2.97%	-2.10%	3.21%
Prev. 2 dmaps	24.55%	6.12%	12.88%	8.19%
Prev. damp + JPL VO	35.71%	24.28%	28.97%	20.98%
Prev. dmap + true VO	36.10%	24.04%	28.04%	21.00%

Run times

Run times for

Intel Core2Quad @ 2.4 GHz,

Fixed local support

Image size	Number of disparities	Iterations		
		20	40	80
512x384	16	**0.315s**	**0.505s**	**0.855s**
512x384	64	**1.257s**	**2.003s**	**3.644s**
512x384	128	**2.786s**	**4.148s**	**7.164s**
1024x768	16	**1.295s**	**2.002s**	**3.389s**
1024x768	64	**4.728s**	**7.654s**	**13.457s**
1024x768	128	**9.507s**	**15.302s**	**26.954s**

- Adaptive local support to preserve fine object structures generates only linear overhead
- Iteration can stop when time is critical, in every iteration step, a momentary optimized disparity map is available
- Potential of significant benefit from parallel implementation on graphics card (CUDA)

Overview

- Two view geometry and 3D scene reconstruction
- Stereo Vision
- Single Camera Structure from Motion
- Summary & Conclusion

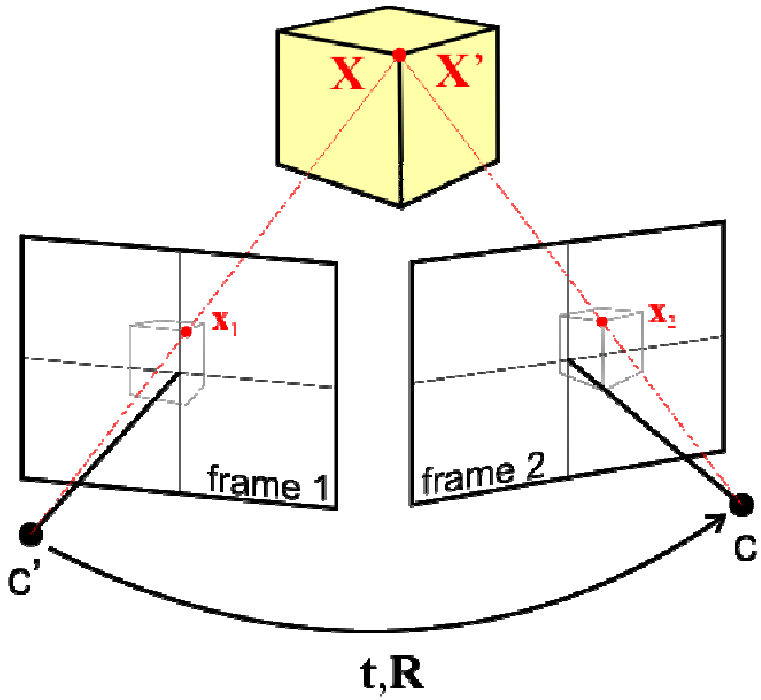
Single moving camera

Single camera, unknown motion

Essential matrix:

$$\mathbf{x}_1^T \mathbf{E} \mathbf{x}_2 = 0 \qquad \mathbf{E} = [\mathbf{t}_\times]\mathbf{R}$$

- **R**, **t** can be recovered up to scale (in theory!)
- 5 independent parameters



Calculation is very noise sensitive !!!
3D distribution of points is important.

Single moving camera

Planar surface

$$aX + bY + cZ + D = 0$$

$$\frac{1}{d}\mathbf{n}^T\mathbf{X} = 1$$

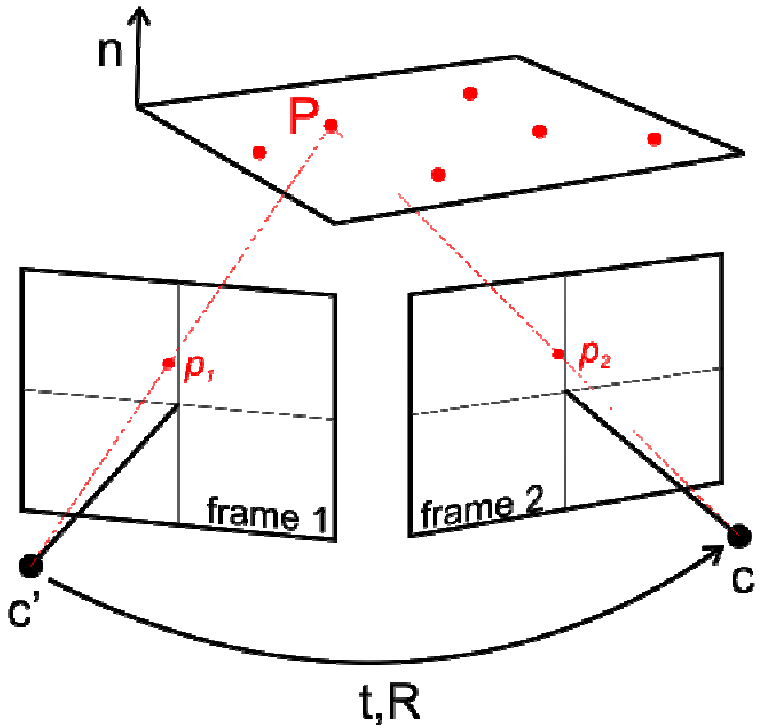
$$\lambda_2\mathbf{x}_2 = \mathbf{R}\lambda_1\mathbf{x}_1 + \mathbf{t}$$

$$\lambda_2\mathbf{x}_2 = (\mathbf{R} + \frac{1}{d}\mathbf{t}\mathbf{n}^T)\lambda_1\mathbf{x}_1$$

$$\mathbf{x}_2 \sim \mathbf{H}\mathbf{x}_1$$

Planar Homography

$$\mathbf{H} = (\mathbf{R} + \frac{1}{d}\mathbf{t}\mathbf{n}^T)$$

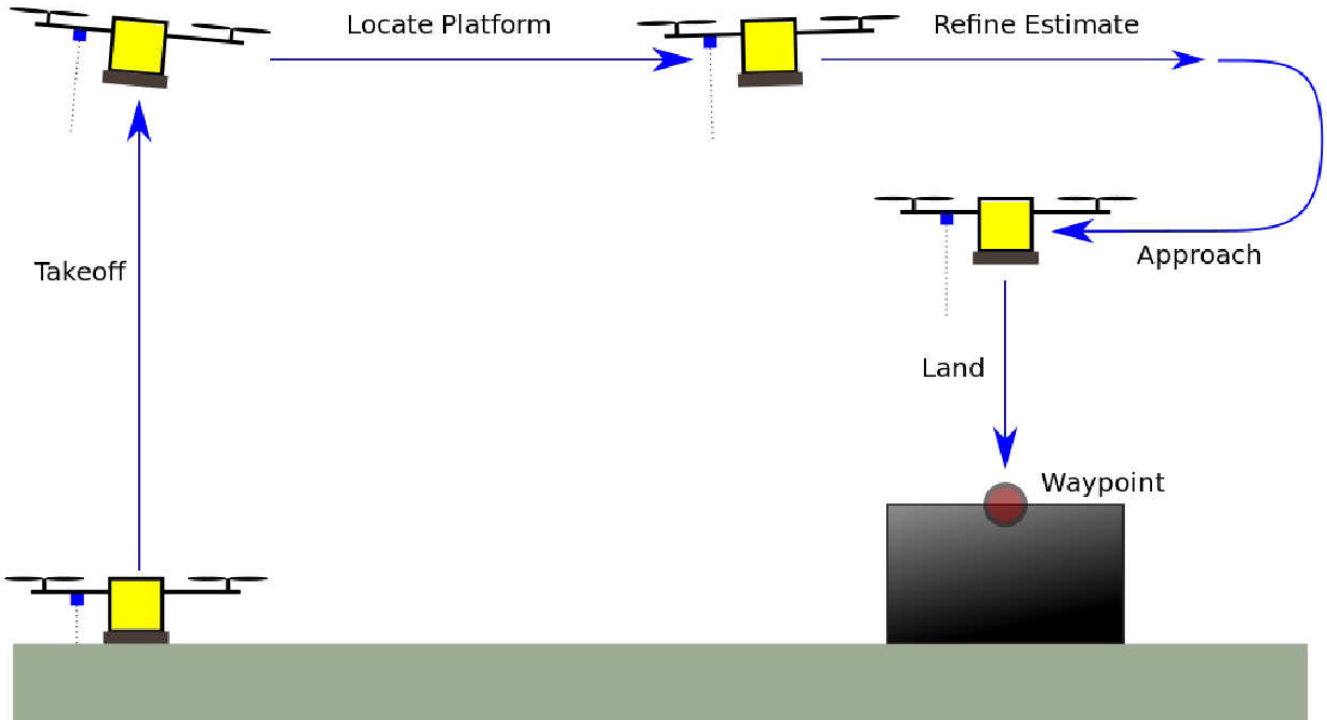


- **R**, **t** & **n** can be recovered (up to scale) [Longuet-Higgins 1986]
- 5 independent parameters

Robust to noise!

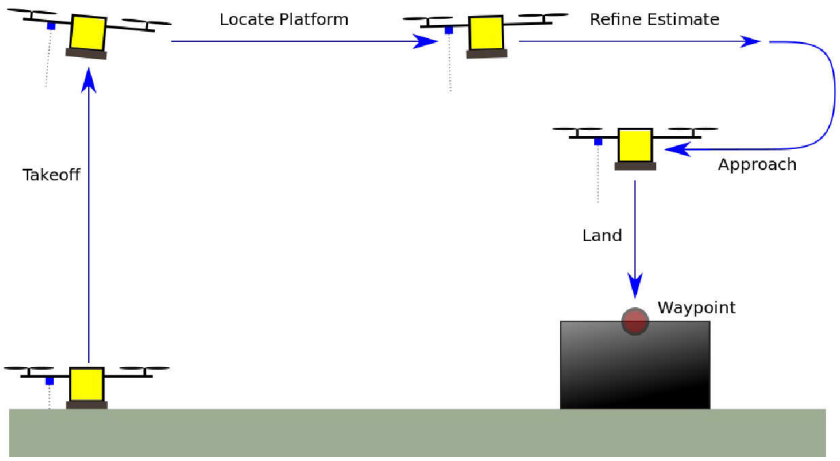
Example: Landing spot detection

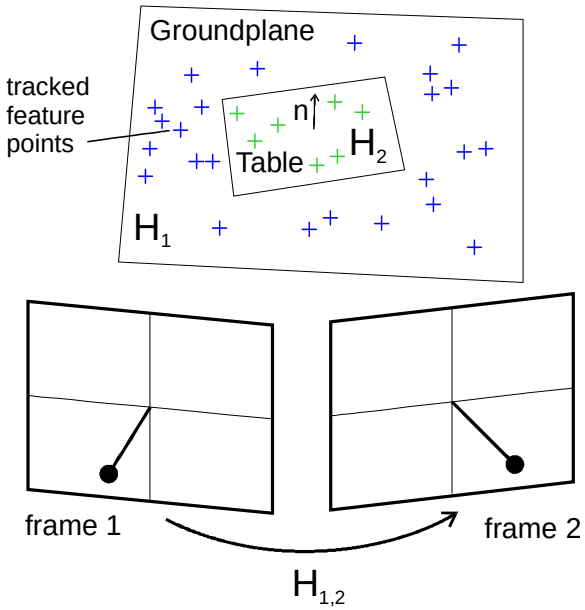
Flying quadrotor with down looking camera



Example: Landing spot detection

Flying quadrotor with down looking camera





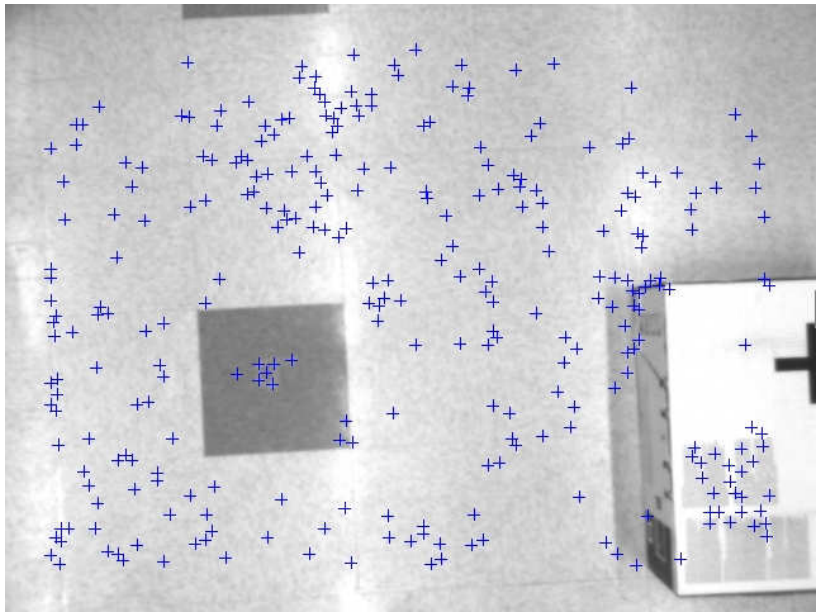
Surface reconstruction
with multiple homographies

$$\mathbf{H}_i \rightarrow \mathbf{R}, \mathbf{t}, \mathbf{n}_i$$

Example: Landing spot detection

Surface detection algorithm

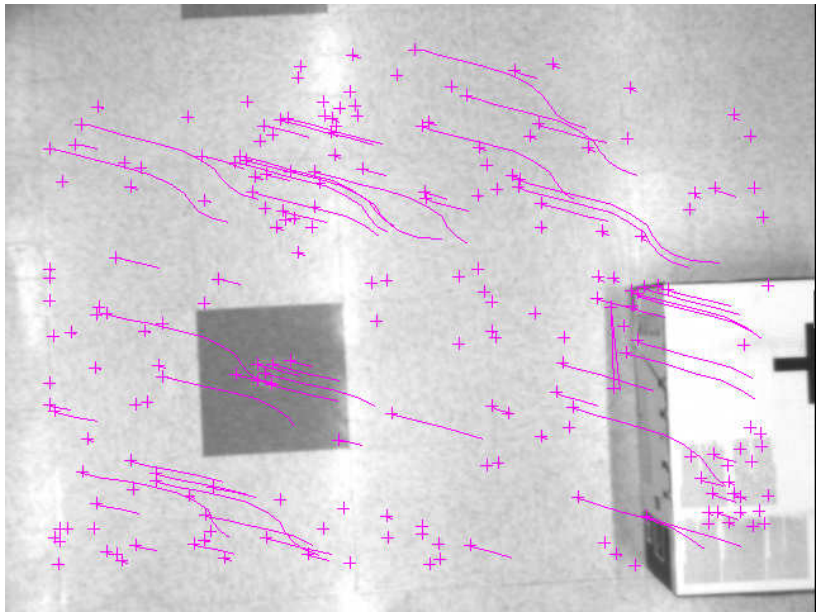
- Detect feature points in images



Example: Landing spot detection

Surface detection algorithm

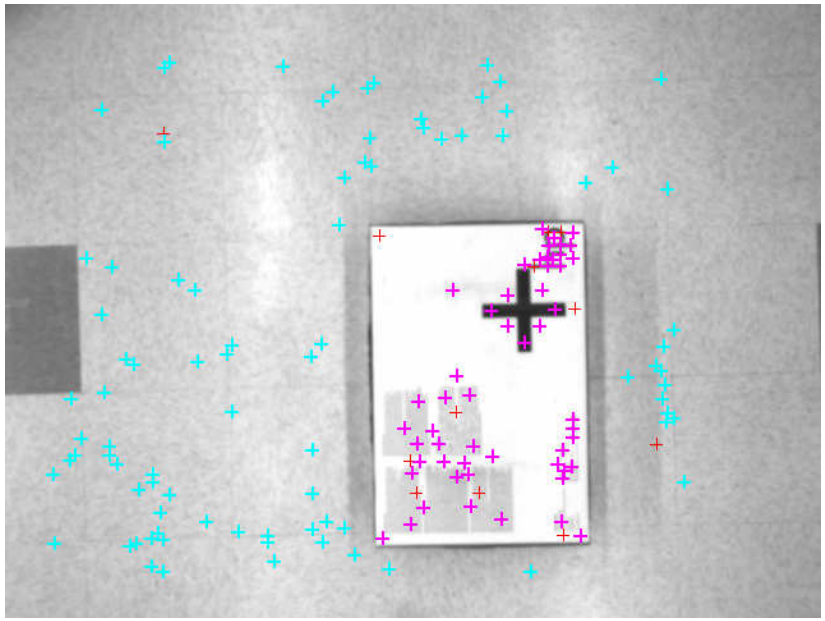
- Detect feature points in images
- Match features between frames and track in image sequence



Example: Landing spot detection

Surface detection algorithm

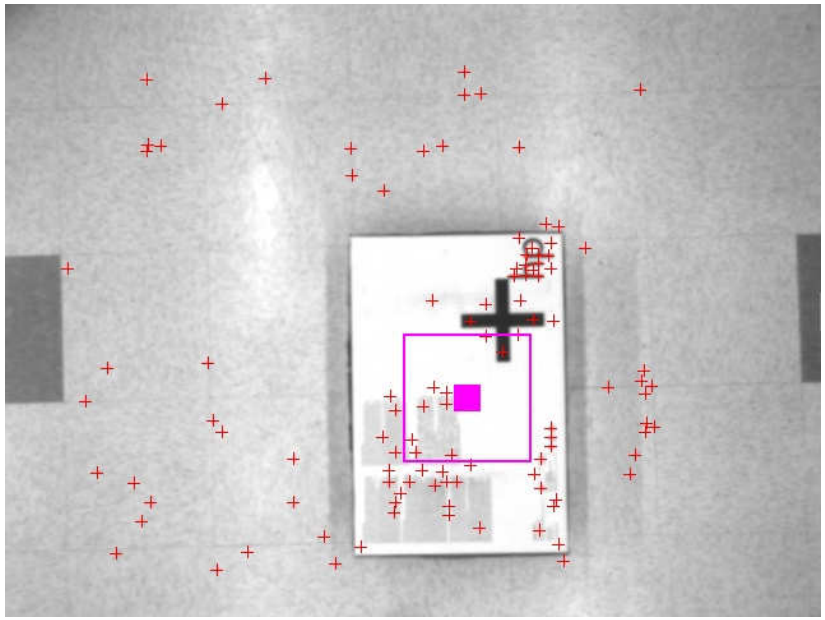
- Detect feature points in images
- Match features between frames and track in image sequence
- Detect planar surface patches by fitting homography to matched features
- Decompose homography into motion parameters $\mathbf{R}$,$\mathbf{t}$ and $\mathbf{n}_i$ defined up to scale



Example: Landing spot detection

Surface detection algorithm

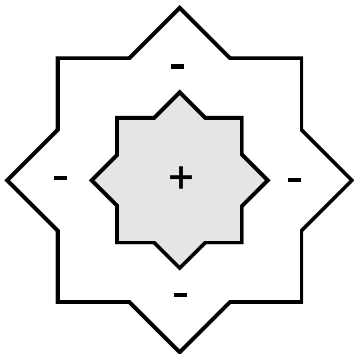
- Detect feature points in images
- Match features between frames and track in image sequence
- Detect planar surface patches by fitting homography to matched features
- Decompose homography into motion parameters $\mathbf{R}$,$\mathbf{t}$ and $\mathbf{n}_i$ defined up to scale
- If elevated surface is found, calculate landing spot

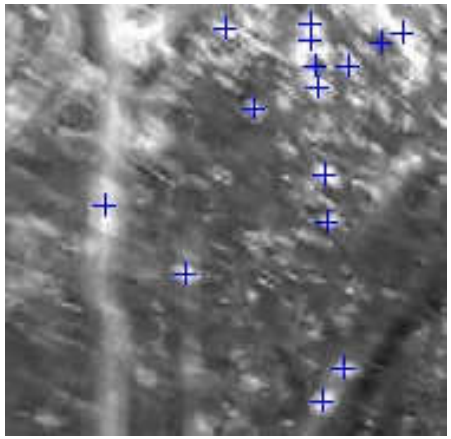


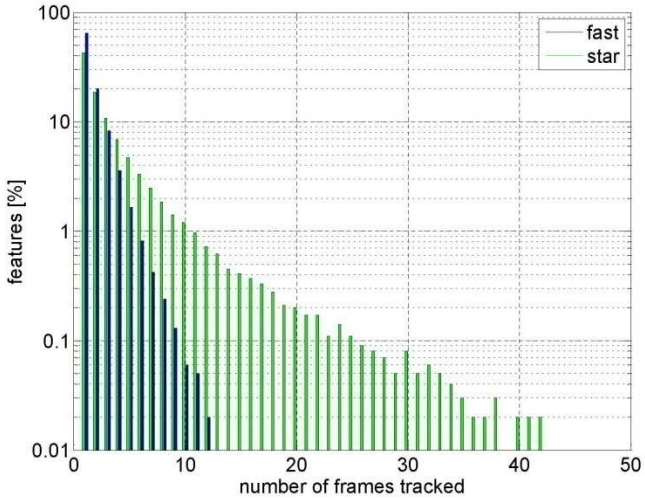
Example: Landing spot detection

Feature detector

- Detector: Center-surround STAR feature detector
  - Detects “blob”-like structures
  - Scale space implementation
  - Longer continuous tracking than corner features in natural scenes
- Feature matching with SURF descriptors



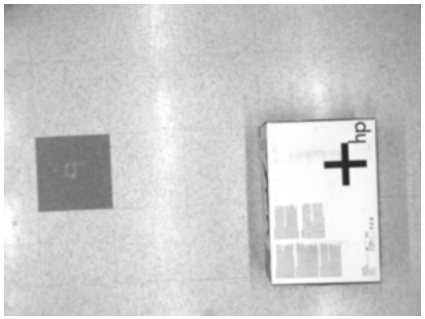


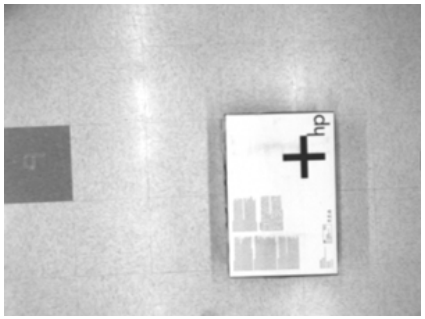


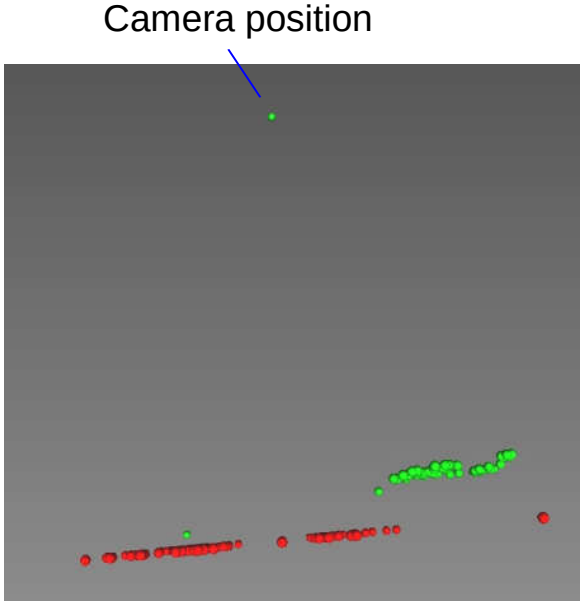
Example: Landing spot detection

3D Reconstruction

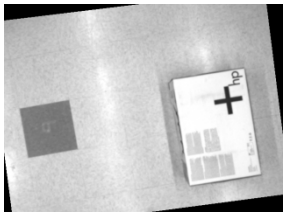
- Point wise reconstruction up to scale using feature matching, multiple homographies, and the plane equation $\mathbf{n}_i^T \mathbf{X} = 1$

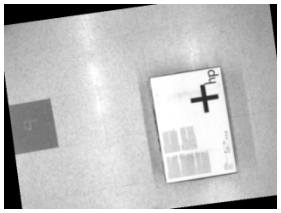


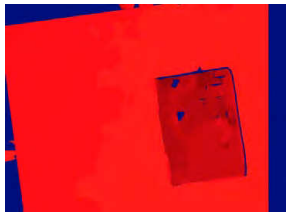




- Dense reconstruction up to scale using stereo algorithm after image alignment

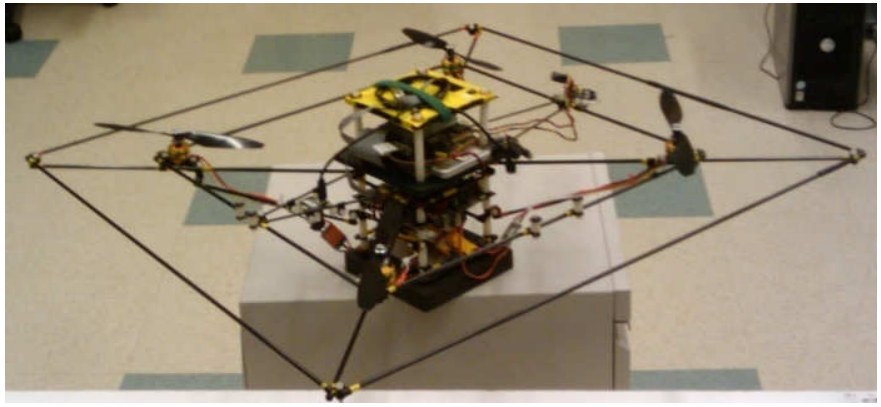






MAST experiment

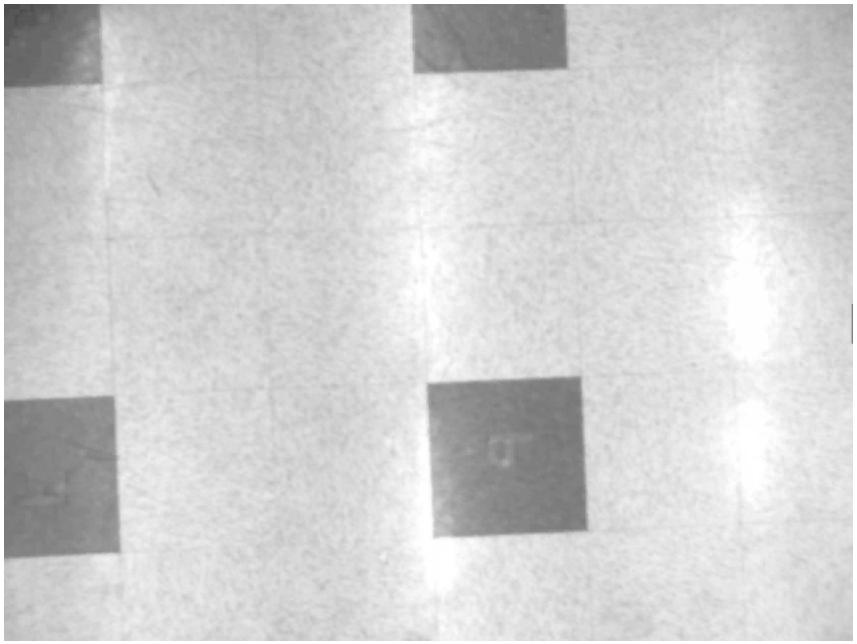
STARMAC quadrotor platform (UC Berkeley)



- ~1.8 kg, 1x1x0.4 m
- 10-20 min flight time
- two CPU variants
  – PC104 (x86)
  – Gumstix Verdex (PXA270)
- VICON system for true state
- 640x480 pixel Point Grey Firefly Camera

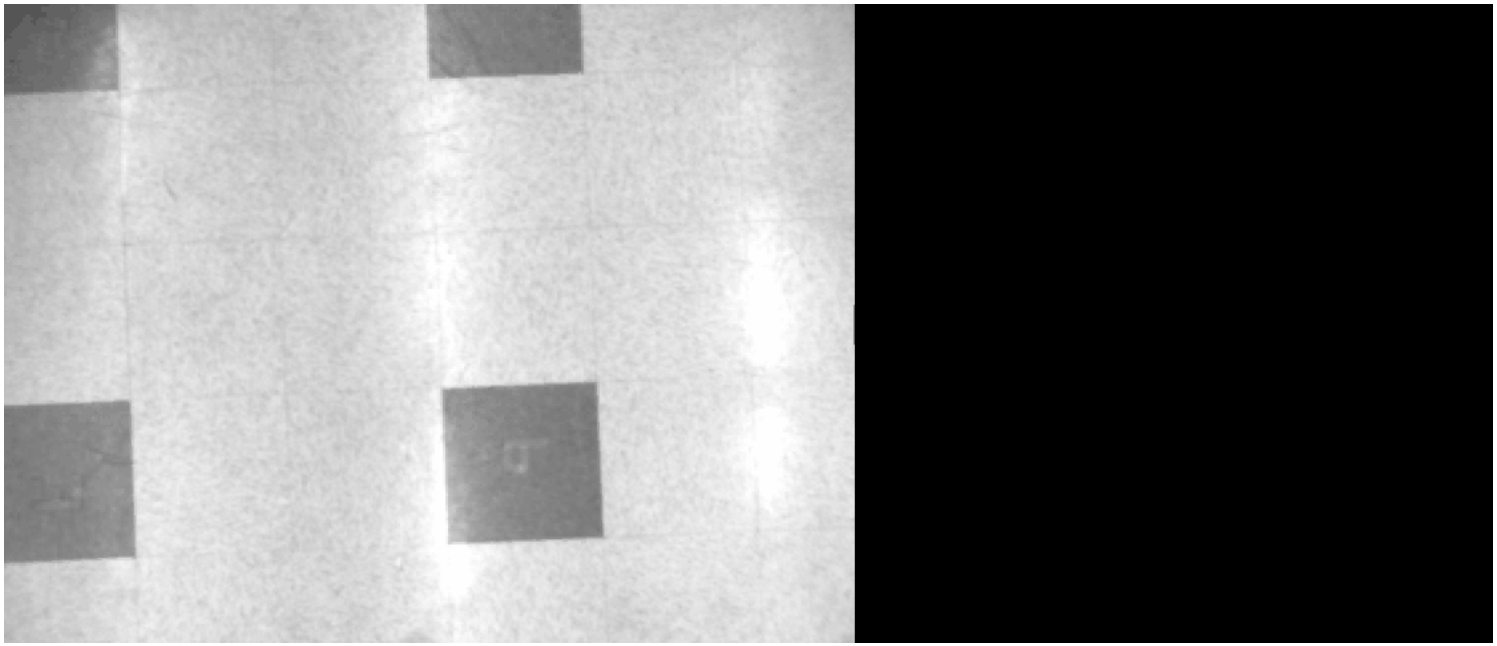
MAST landing experiment

Landing spot detection for quadrotor with downward looking camera



MAST landing experiment

Landing spot detection for quadrotor with downward looking camera

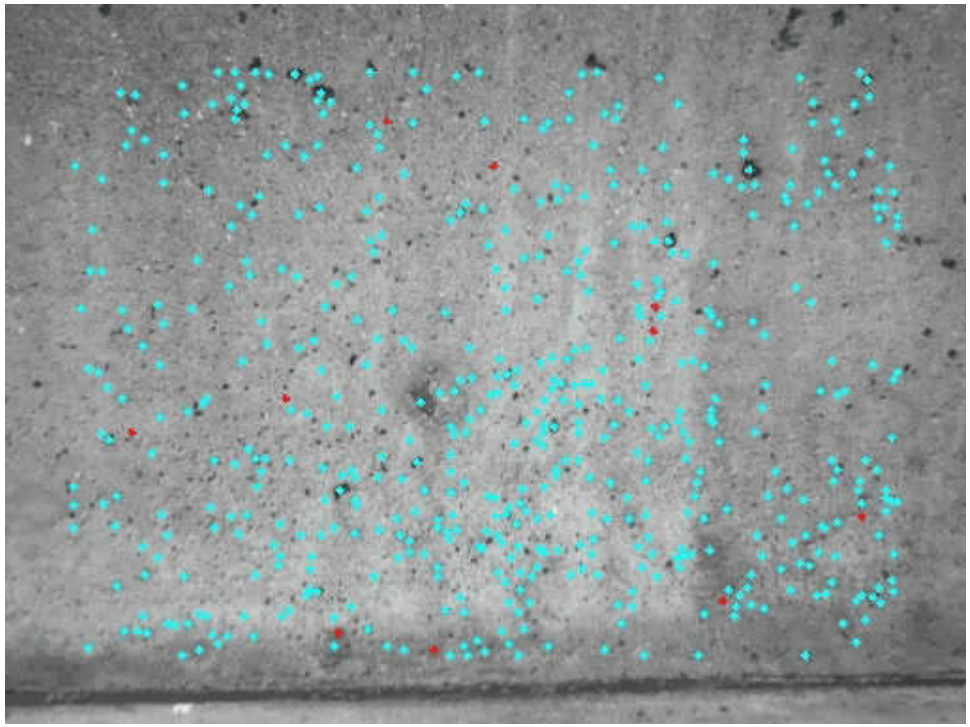


MAST ingress experiment

Door/window detection for quadrotor with forward looking camera

MAST ingress experiment

Door/window detection for quadrotor with forward looking camera



Run times

Calculation times for feature matching & homography calculation

- 800x600 image, 500 features, 3 scales
  Core2Duo 2.4GHz, 1 core only
- Non optimized code

Feature Detection	Integral images	5 ms
	Filtering	30 ms
	Extraction	18 ms
	Descriptor	21 ms
Matching	Upright SURF	14 ms
Homography calculation	2 planes	7 ms
Total		95 ms / 10 fps

Overview

- Two view geometry and 3D scene reconstruction
- Stereo Vision
- Single Camera Structure from Motion
- Summary & Conclusion

Summary & Conclusions

Costrelax dense stereo algorithm

- Very robust to noise
- Low textured areas take advantage from highly textured neighboring areas
- Produces very dense stereo results
- Due to the strict local design, high potential for major speed-up on graphics card implementation with e.g. CUDA.

Single camera Structure-from-motion algorithm

- Longer tracking with Blob like features
- Multiple homography motion estimation
- In future implementation, recover scale from IMU -> visually aided inertial navigation